UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/863,422	05/24/2001	William D. Norcott	50277-1006	9402

7590 07/26/2005

DITTHAVONG & CARLSON, P.C.
10507 Braddock Rd Suite A
Fairfax, VA 22032

EXAMINER	
ALI, MOHAMMAD	

ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 07/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/863,422	NORCOTT, WILLIAM D.	
	Examiner	Art Unit	
	Mohammad Ali	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 February 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-17* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *2/9/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This communication is responsive to the Amendments filed on 2/9/04

Claims 1-17 are pending in this Office Action. Claims 1, 3, 7, and 10 have amended and 12-17 added as a new.

Claim Objections

2. Claims 1-17 objected to because of the following informalities: Technological art is not recites in the claims. Examiner suggests technological art should be added in the claims (e.g. computer-implemented method or system).

Response to Arguments

3. After further search and a thorough examination of the present application, claims 1-17 remain rejected.

Applicant's arguments with respect to claims 1-17 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-17 rejected under 35 U.S.C. 103(a) as being unpatentable over Lorie et al. ('Lorie' hereinafter), US Patent 5,280,612 in view of Robert David Goldring ('Goldring' hereinafter), US Patent 5,553,279.

With respect to claim 1,

Lorie teaches a method for synchronous change data capture (see col. 15, lines 47-45), comprising the steps of:

generating a transaction identifier that uniquely identifies a transaction (a record data pointer, PTR, which points to the location in storage containing the record data for this record version and a transaction identifier TRN, which indicates the sequential identifier for the transaction that created this record version, see col. 8, lines 59-63);

for each operation in a transaction (see col. 13, lines 9-11, Lorie), recording change data for the operation and the transaction identifier in a first database object (the system maintain a record key structure for each record in the database. The database has series of records, each of which is identified by such a key, which can be a logical key or any other suitable type of record identifier, see col. 8, lines 37-46, Fig. 2, Lorie); and

during a commit of the transaction (see col. 11, lines 11-16, Lorie), recording the transaction identifier and a system change number in a second database object,.... (a first version, PTR(1), representing the uncommitted state of record, a second version, PTR(2), representing the last committed value that is not in the stable state, and third

version, PTR(3), representing the stable state for type Q queries. These three versions are organized in a record key structure, see col. 8, lines 50-58 et seq, Lorie).

Lorie does not explicitly indicate the claimed "commit that is later than a previously committed".

Goldring discloses the claimed commit that is later than a previously committed (a computer processing system that receives sequences of updates to source data tables in a data base and records them into an activity log for later retrieval, generates a consistent change data table from the retrieved activity log such that the consistent change data table contains sufficient change information to refresh copies of the source data through multiple generations of target copies by consulting the consistent change data table and applying the table entries to the last prior refreshed source table. The consistent change data table contains committed change operations retrieved from the activity log in the order in which they were committed, beginning with a time no earlier than the last prior refresh (see col. 2, lines 66 to col. 3, lines 11, Goldring).

It would have obvious to one ordinary skill in the data processing art at the time of the present invention, to combine the teachings of the cited references, because commit that is later than a previously committed of Goldring's teaching would have allowed Lorie's system the to produce multi-generational copies of data base tables for replication from one copy level to any other subsequent level, or iteration of copy without losing any change information, as suggested by Goldring, at col. 3, lines 12-14. Commit that is later than a previously committed as taught by Goldring improves consistent change data tables that contains commit time information for placing

transactions in the order which they are committed in the operation (see col. 3, lines 29-34, Goldring).

As to claim 2,

Lorie teaches further comprising the step of: recording an identifier to identify a relative ordering of each operation in the transaction (each new transaction is initiated a TRN serial number is assigned in sequence, for TRN numbers M and following. The UL and NSUL are organized as bit map vectors indexed according to the TRN serial number such that a bit in each list is set to one to include the corresponding TRN serial number on the list, see col. 9, lines 62-67 and col. 8, lines 37-42, Fig. 3).

As to claim 3,

Lorie teaches further comprising, during the commit of the transaction (see col. 8, lines 50-58), the steps of:

obtaining a concurrency lock (query must access the most recent database version, it is relabeled and restarted as an updating transaction and thereby acquire the necessary read locks on all data, see co. 5, lines 4-7 and col. 6, lines 13-15);

after obtaining the concurrency lock, generating the system change number (see col. 5, lines 8-14, Lorie) and performing said recording the transaction identifier and the system change number in the second database object (version control and tracking is implemented by maintaining several version transaction identification lists. In main memory, the system maintains a list of uncommitted update transaction and list of committed but not yet stable state update transactions. The record key version provides a each record version and identifies the creating transaction for each record version,

see col. 5, lines 34-44 and col. 13, lines 10-17 et seq, Lorie), and concluding the commit (a new version of a record is created whenever any updating transaction writes new data to a record that was created by a previous committed transaction, see col. 5, lines 23-25, Lorie); and

after said recording the transaction identifier and the system change number in the second database object (see col. 5, lines 34-44 and col. 13, lines 10-17 et seq, Lorie), releasing the concurrency lock (force to write the commit record log, remove the transaction number form UL and release all locks, see col. 14, lines 51-53 and col. 11, lines 11-15, Lorie).

As to claim 4,

Lorie teaches wherein the first database object comprises a change table and the second database object comprises a transaction table (when transaction activity is low, if a page is modified, then all records on the page can be scanned for supercilious versions, taking into account the UL and NSUL tables "first and second table" as for an update operation. When the transaction activity is very low, a garbage collection transaction can clean up a few pages at a time. This would bring down the average number of record versions in the database and would be particularly appropriate for a dedicated back-end database machine, see col. 13, lines 9-17).

As to claim 5,

Lorie teaches further comprising the step of: associating the change data in the first database object with the system change number in the second database object based on the transaction identifier (version block record key structure is associated with

record data. Version control and tracking is implemented by maintaining several version transaction identification lists. In main memory, the system maintains a list of uncommitted update transaction and list of committed but not yet stable state update transactions. The record key version provides a each record version and identifies the creating transaction for each record version, see col. 5, lines 32-52 and col. 13, lines 10-17 et seq, Lorie).

As to claim 6,

Lorie teaches a computer-readable medium (see col. 5, lines 36-38, Lorie) bearing instructions for synchronous change data capture (see col. 15, lines 47-51, Lorie), said instructions arranged, upon execution (two version rules for actions taken at various stage of the execution of a transaction and a query, see col. 10, lines 18-21 Lorie), to cause one or more processors to perform the steps of the method (when transaction activity is low, if a page is modified, then all records on the page can be scanned for supercilious versions, taking into account the UL and NSUL tables as for an update operation. When the transaction activity is very low, a garbage collection transaction can clean up a few pages at a time. This would bring down the average number of record versions in the database and would be particularly appropriate for a dedicated back-end database machine, see col. 13, lines 9-17, Lorie).

With respect to claim 7,

Lorie teaches a method for processing synchronously captured change data (see col. 15, lines 47-51, Lorie), comprising:

accessing a first database object (access the most recent database version, it relabeled and restarted as an updating transaction and thereby acquire the necessary read locks on all data, see col. 5, lines 3-7, Lorie) comprising change data for an operation performed within a transaction and a transaction identifier that uniquely identifies the transaction (a record data pointer, PTR, which points to the location in storage containing the record data for this record version and a transaction identifier TRN, which indicates the sequential identifier for the transaction that created this record version, see col. 8, lines 59-63, Lorie);

accessing a second database object comprising the transaction identifier and a system change number (version control and tracking is implemented by maintaining several version transaction identification lists. In main memory, the system maintains a list of uncommitted update transaction and list of committed but not yet stable state update transactions. The record key version provides a each record version and identifies the creating transaction for each record version, see col. 5, lines 34-44 and col. 13, lines 10-17 et seq, Lorie); and

associating the change data in the first database object with the system change number in the second database object based on the transaction identifier,.... (version block record key structure is associated with record data. Version control and tracking is implemented by maintaining several version transaction identification lists. In main memory, the system maintains a list of uncommitted update transaction and list of committed but not yet stable state update transactions. The record key version provides

a each record version and identifies the creating transaction for each record version, see col. 5, lines 32-52 and col. 13, lines 10-17 et seq, Lorie).

Lorie does not explicitly indicate the claimed "commit that is later than a previously committed".

Goldring discloses the claimed commit that is later than a previously committed (a computer processing system that receives sequences of updates to source data tables in a data base and records them into an activity log for later retrieval, generates a consistent change data table from the retrieved activity log such that the consistent change data table contains sufficient change information to refresh copies of the source data through multiple generations of target copies by consulting the consistent change data table and applying the table entries to the last prior refreshed source table. The consistent change data table contains committed change operations retrieved from the activity log in the order in which they were committed, beginning with a time no earlier than the last prior refresh (see col. 2, lines 66 to col. 3, lines 11, Goldring).

It would have obvious to one ordinary skill in the data processing art at the time of the present invention, to combine the teachings of the cited references, because commit that is later than a previously committed of Goldring's teaching would have allowed Lorie's system the to produce multi-generational copies of data base tables for replication from one copy level to any other subsequent level, or iteration of copy without losing any change information, as suggested by Goldring, at col. 3, lines 12-14. Commit that is later than a previously committed as taught by Goldring improves consistent change data tables that contains commit time information for placing

transactions in the order which they are committed in the operation (see col. 3, lines 29-34, Goldring).

As to claim 9,

Lorie teaches a computer-readable medium (see col. 5, lines 36-38, Lorie) bearing instructions for synchronous change data capture (see col. 15, lines 47-51, Lorie), said instructions arranged, upon execution (two version rules for actions taken at various stage of the execution of a transaction and a query, see col. 10, lines 18-21 Lorie), to cause one or more processors to perform the steps of the method (when transaction activity is low, if a page is modified, then all records on the page can be scanned for supercilious versions, taking into account the UL and NSUL tables "first and second table" as for an update operation. When the transaction activity is very low, a garbage collection transaction can clean up a few pages at a time. This would bring down the average number of record versions in the database and would be particularly appropriate for a dedicated back-end database machine, see col. 13, lines 9-17, Lorie).

With respect to claim 10,

Lorie teaches a method for synchronous change data capture (see col. 15, lines 47-51, Lorie), comprising the steps of:

generating a transaction identifier that uniquely identifies a transaction (a record data pointer, PTR, which points to the location in storage containing the record data for this record version and a transaction identifier TRN, which indicates the sequential identifier for the transaction that created this record version, see col. 8, lines 59-63);

for each operation in a transaction (see col. 13, lines 9-11, Lorie), recording change data for the operation and the transaction identifier in a change table (the system maintain a record key structure for each record in the database. The database has series of records, each of which is identified by such a key, which can be a logical key or any other suitable type of record identifier, see col. 8, lines 37-46, Fig. 2); and

during a commit of the transaction (see col. 8, lines 50-58), performing the steps of:

obtaining a concurrency lock (query must access the most recent database version, it is relabeled and restarted as an updating transaction and thereby acquire the necessary read locks on all data, see co. 5, lines 4-7 and col. 6, lines 13-15);

after obtaining the concurrency lock, generating a system change number,... (see col. 5, lines 8-14, Lorie) and recording the transaction identifier and the system change number in the database table (version control and tracking is implemented by maintaining several version transaction identification lists. In main memory, the system maintains a list of uncommitted update transaction and list of committed but not yet stable state update transactions. The record key version provides a each record version and identifies the creating transaction for each record version, see col. 5, lines 34-44 and col. 13, lines 10-17 et seq, Lorie); and

after said recording the transaction identifier and the system change number in the database table (see col. 5, lines 34-44 and col. 13, lines 10-17 et seq, Lorie), releasing the concurrency lock (force to write the commit record log, remove the

transaction number form UL and release all locks, see col. 14, lines 51-53 and col. 11, lines 11-15, Lorie).

Lorie does not explicitly indicate the claimed "commit that is later than a previously committed".

Goldring discloses the claimed commit that is later than a previously committed (a computer processing system that receives sequences of updates to source data tables in a data base and records them into an activity log for later retrieval, generates a consistent change data table from the retrieved activity log such that the consistent change data table contains sufficient change information to refresh copies of the source data through multiple generations of target copies by consulting the consistent change data table and applying the table entries to the last prior refreshed source table. The consistent change data table contains committed change operations retrieved from the activity log in the order in which they were committed, beginning with a time no earlier than the last prior refresh (see col. 2, lines 66 to col. 3, lines 11, Goldring).

It would have obvious to one ordinary skill in the data processing art at the time of the present invention, to combine the teachings of the cited references, because commit that is later than a previously committed of Goldring's teaching would have allowed Lorie's system the to produce multi-generational copies of data base tables for replication from one copy level to any other subsequent level, or iteration of copy without losing any change information, as suggested by Goldring, at col. 3, lines 12-14. Commit that is later than a previously committed as taught by Goldring improves consistent change data tables that contains commit time information for placing

transactions in the order which they are committed in the operation (see col. 3, lines 29-34, Goldring).

As to claim 11,

Lorie teaches a computer-readable medium (see col. 5, lines 36-38, Lorie) bearing instructions for synchronous change data capture, said instructions arranged, upon execution (two version rules for actions taken at various stage of the execution of a transaction and a query, see col. 10, lines 18-21 Lorie), to cause one or more processors to perform the steps of the method (when transaction activity is low, if a page is modified, then all records on the page can be scanned for supercilious versions, taking into account the UL and NSUL tables "first and second table" as for an update operation. When the transaction activity is very low, a garbage collection transaction can clean up a few pages at a time. This would bring down the average number of record versions in the database and would be particularly appropriate for a dedicated back-end database machine, see col. 13, lines 9-17, Lorie).

As to claim 8,

Lorie teaches wherein the step of associating includes performing a database operation on the first database object and the second database object (when transaction activity is low, if a page is modified, then all records on the page can be scanned for supercilious versions, taking into account the UL and NSUL tables "first and second table" as for an update operation. When the transaction activity is very low, a garbage collection transaction can clean up a few pages at a time. This would bring down the average number of record versions in the database and would be particularly

appropriate for a dedicated back-end database machine, see col. 13, lines 9-17 and col. 5, lines 48-52).

Lorie does not explicitly indicate the claimed "join operation".

Goldring discloses the claimed join operation (the Consistent_Change_Data table includes only updates that have been committed and is created by performing an SQL join operation on the Change_Data and VOW tables, see col. 7, lines 1-3, Fig. 6, Goldring).

It would have obvious to one ordinary skill in the data processing art at the time of the present invention, to combine the teachings of the cited references, because join operation of Goldring's teaching would have allowed Lorie's system the to produce multi-generational copies of data base tables for replication from one copy level to any other subsequent level, or iteration of copy without losing any change information, as suggested by Goldring, at col. 3, lines 12-14. Join operation as taught by Goldring improves consistent change data tables that contains commit time information for placing transactions in the order which they are committed in the operation (see col. 3, lines 29-34, Goldring).

As to claim 12,

Lorie teaches, wherein the system change number indicates an event occurring between said obtaining the concurrency lock and said releasing the concurrency lock (see col. 5, lines 8-14, Lorie).

As to claim 13,

Lorie teaches wherein the system change number indicates an event occurring before the commit of the transaction (query must access the most recent database version, it is relabeled and restarted as an updating transaction and thereby acquire the necessary read locks on all data, see co. 5, lines 4-7 and col. 6, lines 13-15 et seq.).

As to claim 14,

Lorie teaches generating a commit system change number for the transaction (see col. 5, lines 34-44 and col. 13, lines 10-17 et seq, Lorie) that is later then the system change number (force to write the commit record log, remove the transaction number form UL and release all locks, see col. 14, lines 51-53 and col. 11, lines 11-15, Lorie).

As to claim 15,

Lorie teaches wherein the system change number indicates an event occurring between said obtaining the concurrency lock and said releasing the concurrency lock (see col. 5, lines 8-14 et seq., Lorie).

As to claim 16,

Lorie teaches wherein the system change number indicates an event occurring before the commit of the transaction (version control and tracking is implemented by maintaining several version transaction identification lists. In main memory, the system maintains a list of uncommitted update transaction and list of committed but not yet stable state update transactions. The record key version provides a each record version and identifies the creating transaction for each record version, see col. 5, lines 8-14, 34-44 and col. 13, lines 10-17 et seq, Lorie).

As to claim 17,

Lorie teaches generating a commit system change number for the transaction that is later then the system change number (version control and tracking is implemented by maintaining several version transaction identification lists. In main memory, the system maintains a list of uncommitted update transaction and list of committed but not yet stable state update transactions. The record key version provides a each record version and identifies the creating transaction for each record version, see col. 5, lines 34-44 and col. 13, lines 10-17 et seq, Lorie).

Remarks

6. In response to the applicant's arguments "indicates a timing of the commit that is later than a previously committed transaction".

The Examiner respectfully submits that Lorie and Goldring discloses the particular limitation as stated in the detail Office Action.

In response to the applicant's arguments "wherein the system change number is recorded during a commit of the transaction and indicates a timing of the commit that is later than a previously committed transaction".

The Examiner respectfully submits that Lorie and Goldring discloses the particular limitation as stated in the detail Office Action.

Hence, Applicants' arguments do not distinguish over the claimed invention over the prior art of records.

In light of the foregoing arguments, the 103 rejections are hereby sustained.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mohammad Ali whose telephone number is (571) 272-4105. The examiner can normally be reached on Monday-Thursday (7:30 am-6:00 pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene can be reached on (571) 272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-272-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mohammad Ali
Primary Examiner
Art Unit 2167

MA
July 23, 2005